



UNIVERSIDAD NACIONAL DE LA PLATA

Informe Final Taller de Proyecto I: Dispensador de alimento de mascotas automático

Grupo 1

Castro Bustos Juan Martín - 02004/9

Graziani Alejandro Martín - 02117/8

Reyes Manuel - 01852/0

Rodríguez Joaquín - 02137/2

Fecha: 21/02/2026

Ingeniería en Computación

Facultad de Ingeniería – Universidad Nacional de La Plata

Avenida 1 750, 1900 La Plata, Buenos Aires, Argentina

ÍNDICE

INTRODUCCIÓN.....	1
OBJETIVOS DEL PROYECTO.....	2
Objetivo General.....	2
Objetivos Primarios.....	2
Objetivos Secundarios.....	3
Descripción Técnico-Conceptual.....	3
REQUERIMIENTOS.....	4
Funcionales.....	4
No funcionales.....	4
DISEÑO DE HARDWARE.....	5
Diseño de la solución propuesta.....	5
Lista de materiales.....	6
Descripción de interfaces eléctricas.....	6
Alimentación del circuito y la EDU-CIAA.....	10
Esquemático del sistema.....	11
PCB.....	12
Diseño.....	12
Representación 3D.....	13
Descripción.....	14
Diseño de hardware final.....	14
Ensamblaje.....	15
Gestión de atascos de alimento en la hélice dispensadora.....	16
DISEÑO DEL FIRMWARE.....	16
Modularización.....	17
1. Módulo de comunicación (ESP32 – WIFI).....	17
2. Módulo de control de dispensación (servomotor).....	17
3. Módulo de sensor de peso (galga extensiometrífica).....	17
4. Módulo de control de tiempo (RTC).....	18
5. Módulo de gestión de datos (APP).....	18
Estructura de archivos.....	19
Aplicación.....	21
Estructura de archivos de la APP:.....	21
Diagrama de flujo APP.....	21
Prototipo final de la APP.....	22
DESARROLLO DE SOFTWARE.....	24
Avances de desarrollo (firmware y conectividad).....	24
Avances de desarrollo (interfaz de usuario y aplicación).....	25
Diseño de software final.....	27
7. ENSAYOS Y MEDICIONES.....	28
8. CONCLUSIONES.....	30
Cumplimiento de Objetivos.....	30
Evaluación de Requerimientos.....	30
Conclusión General.....	31

Trabajos a futuro.....	31
División de tareas final.....	32
Cronograma.....	34
BIBLIOGRAFÍA.....	36
ANEXOS.....	37

INTRODUCCIÓN

En la actualidad, el cuidado de las mascotas ha adquirido un rol fundamental en la vida de las personas, no solo por la compañía que brindan, sino también por la responsabilidad que implica garantizar su bienestar. Una de las necesidades básicas de los animales domésticos es la alimentación, la cual debe administrarse en horarios regulares y en cantidades adecuadas para mantenerse saludable. Sin embargo, en la rutina diaria, muchas veces nos vemos imposibilitados de estar presentes en todo momento, lo que puede derivar en irregularidades en la alimentación. Esta problemática nos motiva a desarrollar un sistema automatizado que permita ayudar a los dueños en el manejo eficiente y seguro de la comida para sus mascotas.

El proyecto propuesto consiste en el diseño y construcción de un dispensador automático de alimento para mascotas, que permita programar la alimentación en intervalos de tiempo configurables.

Los principales desafíos incluyen el diseño de un mecanismo confiable de dispensado que evite bloqueos y la implementación de un sistema de control programable que resulte intuitivo para el usuario. Además, se deberá garantizar la seguridad y durabilidad del dispositivo, dado que estará expuesto al contacto directo con los animales.

La importancia del proyecto radica en brindar una solución práctica para los dueños de mascotas, mejorando su calidad de vida y asegurando una alimentación más regular y controlada para los animales. Además, la experiencia obtenida en el diseño e implementación de este sistema permite aplicar conceptos de electrónica, programación y control de procesos, combinando teoría y práctica en un dispositivo de uso real.

En conclusión, el dispensador automático de comida para mascotas se presenta como una propuesta que responde a una necesidad concreta de la vida cotidiana mediante la aplicación de herramientas tecnológicas.

OBJETIVOS DEL PROYECTO

El proyecto tiene como propósito resolver el problema de la alimentación irregular de las mascotas mediante el desarrollo de un sistema automático de dispensado de comida. Este sistema busca ofrecer una solución flexible que combine la programación de horarios de alimentación con la interacción remota mediante una aplicación móvil, asegurando que el animal reciba alimento de forma controlada y en el momento adecuado.

Objetivo General

Diseñar y construir un prototipo de dispensador automático de comida para mascotas, capaz de operar en dos modos:

1. Modo programado, donde el usuario establece intervalos de tiempo para la liberación de alimento.
2. Modo remoto mediante la aplicación, en el cual el usuario puede configurar parámetros de funcionamiento y dispensar la comida desde su celular.

Objetivos Primarios

Los objetivos que deben cumplirse en tiempo y forma para alcanzar el éxito del proyecto son:

1. Implementar un sistema de control basado en un microcontrolador que gestione los modos de operación.
2. Diseñar e integrar un mecanismo de dispensado confiable, evitando bloqueos y garantizando repetibilidad.
3. Configurar un temporizador para que sea programable, pudiendo satisfacer los horarios dictados por el usuario.
4. Implementar una aplicación móvil que permita:
 - a. Configurar los intervalos y el modo de operación del dispensador
 - b. Activar de forma manual la entrega de alimento mediante un botón visual

Objetivos Secundarios

Como posibles mejoras, sujetas a la disponibilidad de materiales, costos y tiempo de desarrollo, se proponen:

1. Añadir un sistema de registro del consumo de alimento (por ejemplo, cantidad dispensada por día).
2. Integrar sensores adicionales (peso del recipiente, etc.) que permitan mejorar el monitoreo del sistema.

Descripción Técnico-Conceptual

El sistema se compone de diferentes módulos interconectados:

- **Microcontrolador:** microcontrolador encargado de gestionar todo
- **Interfaz de Usuario:** permite al usuario configurar parámetros básicos (tiempo, cantidad, modo de operación).
- **Módulo de Comunicación:** Este módulo se encarga de traducir la comunicación entre la aplicación y el microcontrolador.
- **Módulo de alimentación:** fuente de energía que suministra el voltaje necesario al sistema.
- **Temporizador:** Reloj configurable con distintos horarios para dictar cada cuanto dispensar la comida.
- **Galga extensiométrica:** Indica el peso del depósito del alimento para saber si la comida disponible es suficiente o se debe llenar.
- **Motor/Dispensador:** Encargado de hacer el movimiento para dejar caer el alimento.

En la Fig. 1 se presenta el diagrama en bloques del sistema a desarrollar:

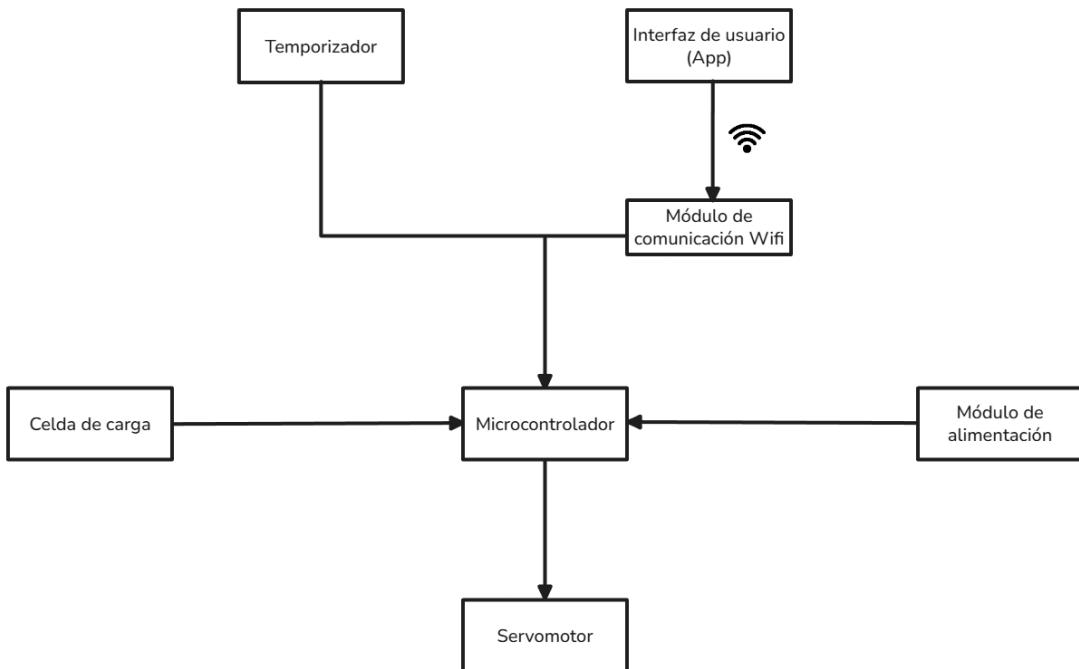


Figura 1. Diagrama en bloques del sistema propuesto.

REQUERIMIENTOS

Funcionales

1. El depósito debe contener alimento y permitir el sensado del peso actual, para calcular la cantidad restante de alimento en gramos y porciones.
2. El dispositivo debe suministrar comida automáticamente cuando se cumpla el horario de alimentación.
3. Ofrecer modos de uso:
 - a. por horario.
 - b. manual, mediante botones físicos.
 - c. cada determinado tiempo.
4. Se deberá poder definir el peso en gramos de las porciones a dispensar.
5. Se deberá poder configurar el funcionamiento del dispositivo en una aplicación móvil.
6. Informar en la aplicación móvil y en una pantalla física la cantidad de comida disponible en el depósito en gramos y cantidad de porciones.

No funcionales

1. Uso del microcontrolador Edu CIAA.

2. El dispositivo deberá tener un tamaño y peso apto para uso doméstico.
3. El diseño debe ser amigable con las mascotas.
4. Los tiempos deben adaptarse al cronograma de la materia.

DISEÑO DE HARDWARE

Diseño de la solución propuesta

El sistema propuesto consiste en un dispensador automático de alimento para mascotas que integra componentes mecánicos, electrónicos y de comunicación, permitiendo tanto el funcionamiento automático como el control manual por parte del usuario.

El dispositivo podrá ser utilizado de dos maneras: de forma remota mediante una aplicación móvil, o de manera manual a través de la interfaz física con pantalla y botones.

En primer lugar, el alimento se deposita en una tolva que actúa como contenedor principal. En la parte inferior de la misma se encuentra una hélice accionada por un servomotor, cuya función es dosificar la cantidad de alimento que se libera en cada operación. La dosificación es configurable mediante cualquiera de las dos interfaces. El diseño contará con una balanza basada en una galga extensiométrica, la cual mide en tiempo real el peso del alimento dentro del depósito, asegurando precisión en la entrega de la ración. Además, la balanza podrá informar al usuario la cantidad de comida disponible en el depósito en cualquier momento.

El procesamiento y control principal del sistema se realiza mediante el microcontrolador LPC4337, encargado de gestionar los sensores, el servomotor y la interfaz local. En paralelo, se emplea un módulo ESP32 para la comunicación inalámbrica, lo que permite establecer el enlace con la aplicación móvil y transmitir información sobre el estado del dispositivo y las dispensaciones realizadas.

El diseño garantiza que se cumplan los requerimientos planteados: control automático de raciones, interacción remota con el usuario, medición precisa de la cantidad de alimento y posibilidad de operación manual.

Lista de materiales

1. 1 microcontrolador LPC4337.
2. 1 servomotor MG995 modificado para que trabaje 360°.
3. 1 microcontrolador ESP8266.
4. 1 sensor de peso HX711 con celda de carga de 5kg.
5. 3 botones mecánicos.
6. 1 display OLED.
7. 1 fuente de alimentación.
8. Soporte estructural impreso en PLA.
9. Pines (todos con separación de 0,1in):
 - a. Dos tiras de 2x20 pines
 - b. Cinco pares de pines
 - c. Dos tiras de 4 pines
 - d. Una tira de 3 pines
 - e. Una tira de 6 pines
 - f. Una tira de 2x4 pines
10. Tuercas y tornillos.
11. 6 tuercas de inserción M3.
12. Placa protoboard.
13. 2 rulemanes R3ZZ.
14. 1 varilla roscada M3.
15. Base de madera.

Presupuesto aproximado = \$70.000

Descripción de interfaces eléctricas

- Servomotor MG995:



Imagen 1. Servomotor MG995

- Tipo de señal: PWM de 50Hz de frecuencia
- Protocolo: PWM
- Alimentación: 4.8V a 7.2V desde fuente externa al microcontrolador
- Se conecta al GPIO8, alimentación de 5V y a GND.

- Microcontrolador ESP8266

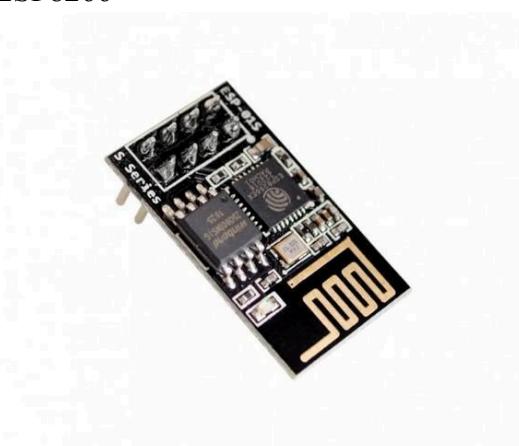


Imagen 2. ESP82cc

- Tipo de señal: UART TTL
 - Protocolo: UART
 - Alimentación: 2.3V a 3.6V
 - Conexión a los pines RS232_TX, RS232_RX, pin1 del P2, GND
- Sensor de peso HX711

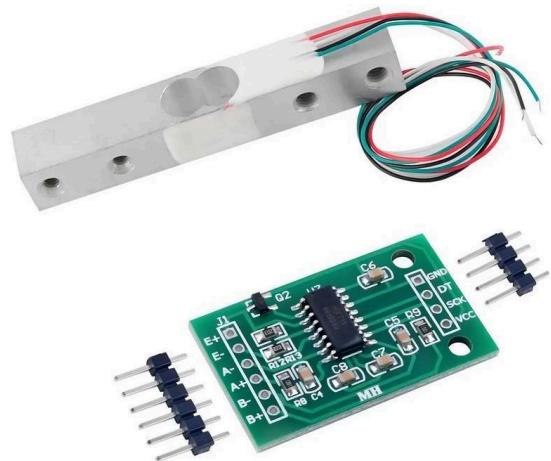


Imagen 3. Galga HX711

- Alimentación: 2.6V a 5.5V
- Conexión SCK al GPIO5, DT al GPIO7, GND, alimentación 5V

- Botones mecánicos



Imagen 4. Botón mecánico

- Tipo de señal: digital.
- Conexiones a los pines: GPIO0, GPIO2, GPIO4.

- Display OLED

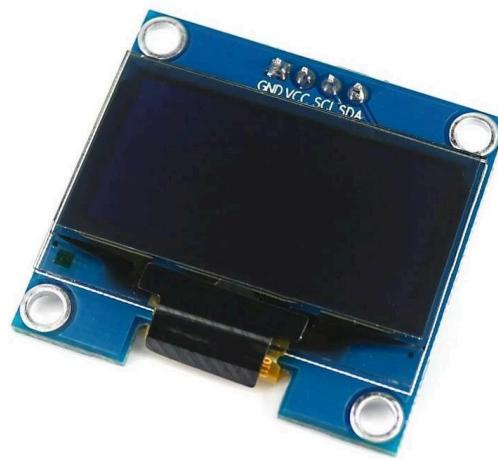


Imagen 5. Display OLED SH1106

- Tipo de señal: digital
- Protocolo: I2C
- Alimentación: 2.4V a 3.6V
- Conexiones: SCL al pin21 de P1 (I2C SCL), SDA al pin19 del P1 (I2C SDA), alimentación mediante fuente externa y GND

Elemento	Función en el Circuito
Microcontrolador LPC4337	Microcontrolador principal que gestiona la lógica del sistema, control del servomotor, lectura de sensores y comunicación con la interfaz local y el ESP8266.
Servomotor MG995	Actuador encargado de realizar el movimiento mecánico de la hélice para dispensar la porción de alimento.
Celda de carga HX711	Módulo sensor de peso del alimento restante en el depósito para calcular la ración y el nivel disponible.
Display OLED SH1106	Interfaz visual que muestra información relevante al usuario, como la cantidad de comida disponible, el modo de operación o la hora de la siguiente dispensación.
Microcontrolador ESP8266/ESP32	Módulo de comunicación inalámbrica (Wi-Fi) que establece la conexión para el control remoto y la transmisión de datos.
Fuente de alimentación externa (5V/1A)	Suministra la energía necesaria al sistema, especialmente al servomotor y otros periféricos, dado que el consumo total excede la capacidad segura de la fuente de la EDU-CIAA.
Botones mecánicos	Interfaz física local que permite la interacción manual del usuario para configuraciones básicas o dispensación manual.

Alimentación del circuito y la EDU-CIAA

A continuación, se listan detalles de consumo de los periféricos:

- EDU-CIAA: promedio de 200mA
- Servomotor MG995: 500mA a 900mA a 6V, pico máximo 2,5A (si se estanca).¹
- Celda de carga HX711: menor a 1.5mA en funcionamiento normal.²
- Display OLED SH1106: 200µA por segmento, pico de 27mA.³
- Microcontrolador ESP32: pico de 500mA.⁴
- Botones: consumo despreciable (en el orden de 1mA)

A partir de los resultados de consumo de los periféricos y actuadores, se determinó que la potencia requerida excede la capacidad de alimentación que puede proporcionar la EDU-CIAA de manera segura. Por esta razón, se decidió incorporar una fuente de alimentación externa.

En concreto, se utilizará una fuente de 5V y 2A dedicada a alimentar todos los componentes, excluyendo el ESP8266, ya que requiere alimentación de hasta 3,6V. Esta configuración permite aislar la carga pesada del servomotor y garantizar la estabilidad del sistema sin sobrecargar la fuente interna del microcontrolador.

¹ <https://www.alldatasheet.com/datasheet-pdf/download/1131873/ETC2/MG996R.html>

² <https://www.alldatasheet.com/datasheet-pdf/view/1132222/AVIA/HX711.html>

³ https://cdn.velleman.eu/downloads/29/infosheets/sh1106_datasheet.pdf

⁴ <https://www.alldatasheet.es/datasheet-pdf/view/1148023/ESPRESSIF/ESP32.html>

Esquemático del sistema

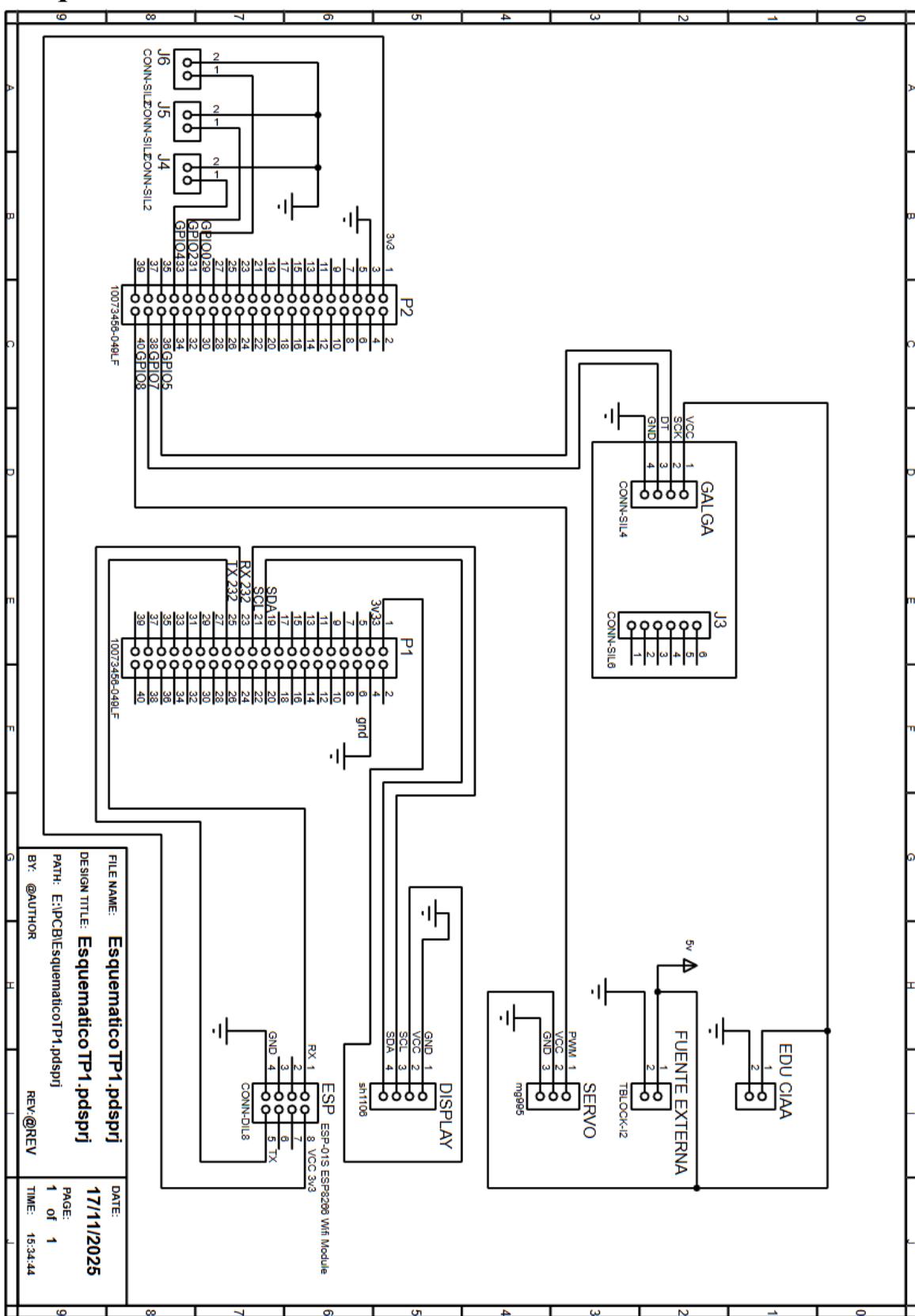


Imagen 6. Esquemático Proteus

PCB

Diseño

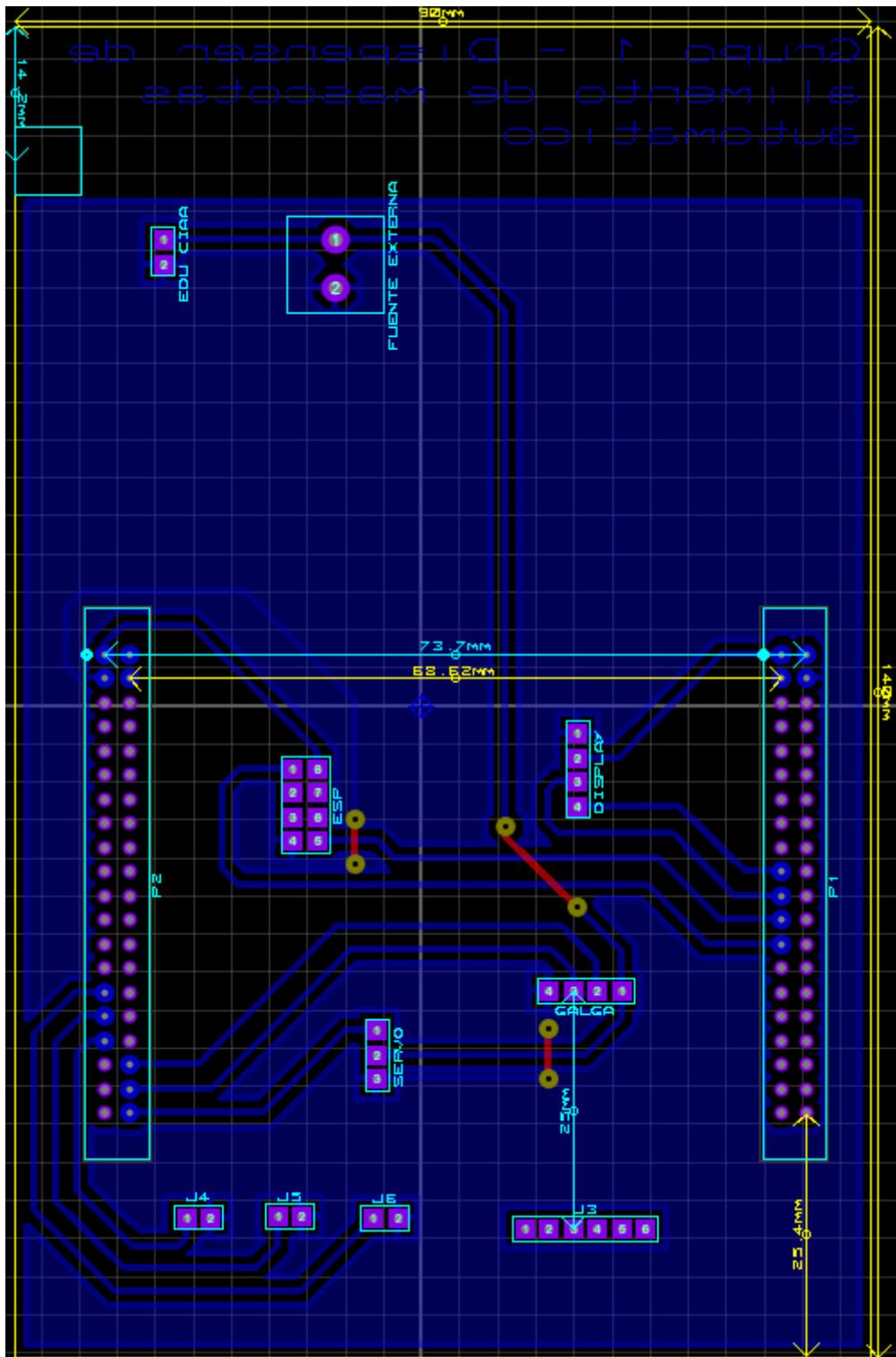


Imagen 7. PCB Proteus

Representación 3D

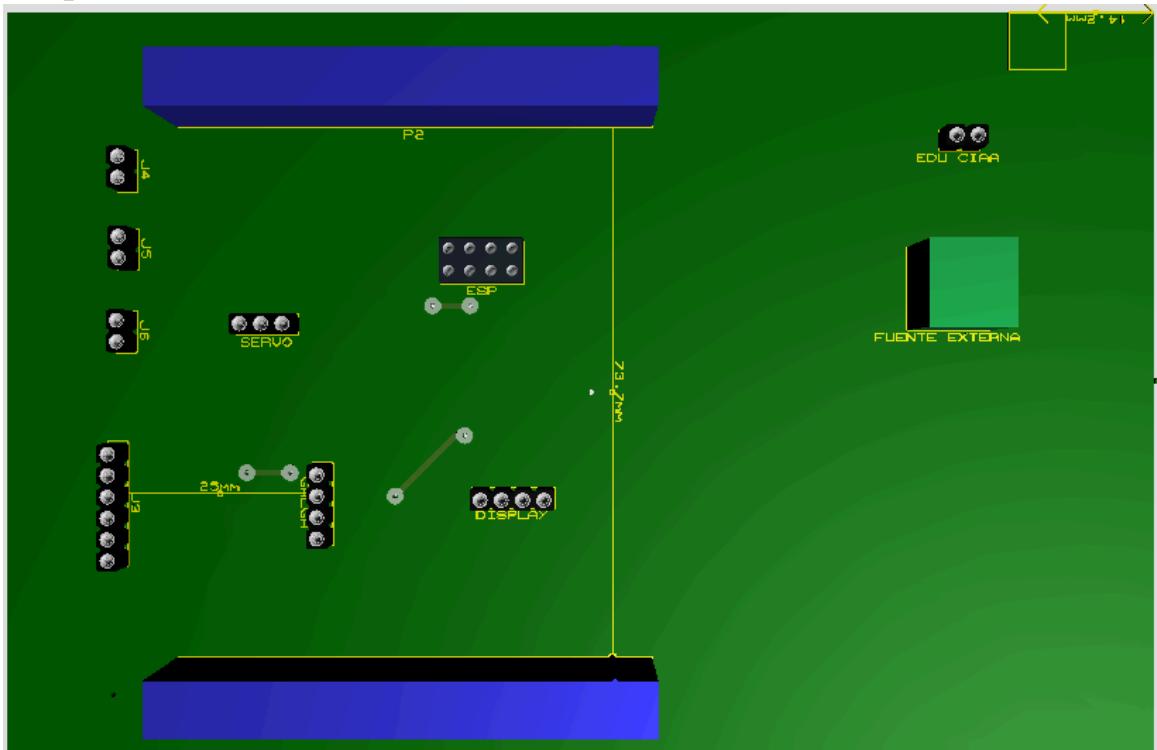


Imagen 8. Representación 3D delantera

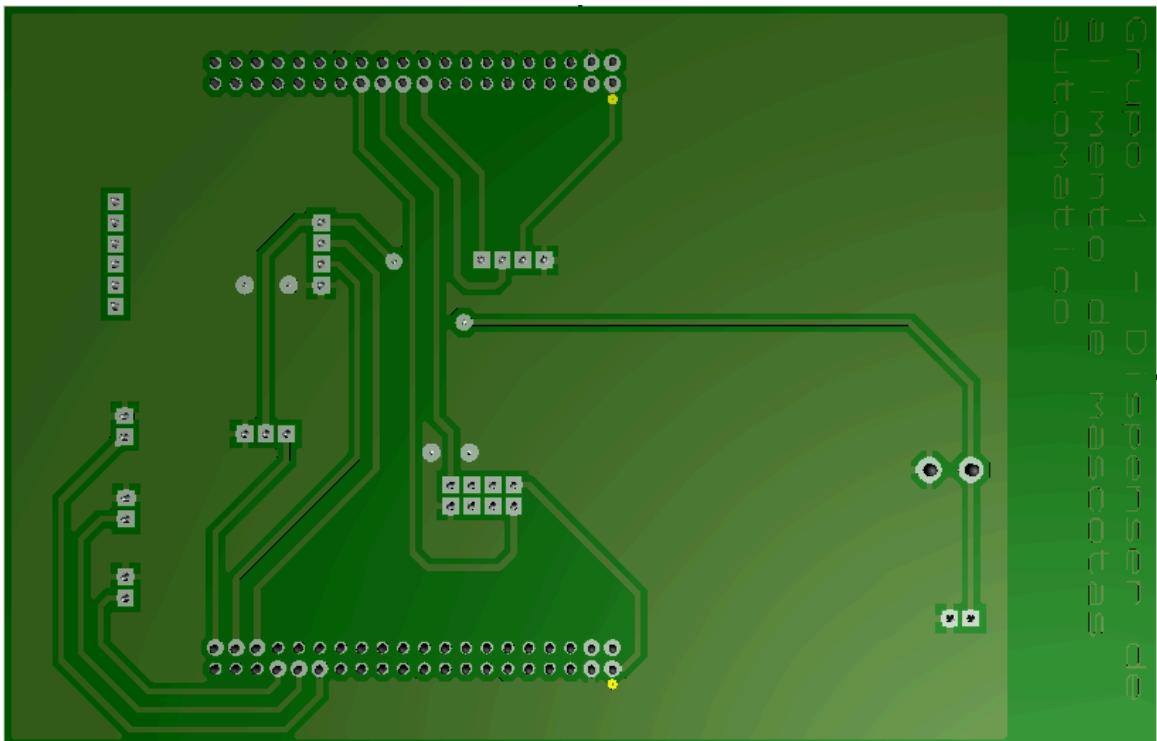


Imagen 9. Representación 3D trasera

GRUPO 4 - Dispensador de
ALIMENTO DE MASCOTAS
AUTOMATICO

Descripción

La placa PCB mostrada anteriormente va a actuar como interfaz entre los periféricos y actuadores y el microcontrolador EDU-CIAA. Tiene unas dimensiones de 90x140x1.6mm. La placa se montará directamente por encima del microcontrolador y dispondrá de borneras de conexión donde se conectarán los elementos para facilitar y ordenar la integración de todas las partes del sistema.

La PCB está compuesta por dichas borneras, las cuales tienen separación entre pines de 0,1in. Las mismas se detallan a continuación:

- Dos tiras de 2x20 columnas de pines, para las conexiones a la EDU-CIAA.
- Bornera con tornillos para fuente externa.
- Cinco pares de pines destinados para:
 - uno para la alimentación del microcontrolador.
 - uno para cada uno de los tres botones.
- Una tira de 4 pines para el display OLED.
- Una tira de 3 pines para el servomotor.
- Para la balanza se utilizaron una tira de 4 pines para conexión al conversor analógico-digital y una tira de 6 pines para soporte de la conexión del conversor a la balanza.
- Una tira de dos columnas de 2x4 pines para el ESP8266.

Se utilizaron tracks T30 de 0,7mm de ancho y pads de 2mm de diámetro y 0,7mm para los agujeros, la separación entre tracks es de mínimo 1mm y 0,7mm hasta el borde de la PCB.

La placa cuenta con una capa dedicada al plano de masa, la cual actúa como referencia a tierra o GND común para todos los dispositivos del circuito. Este plano se extiende por toda la placa, permitiendo las conexiones de todos los periféricos.

En la parte superior de la placa se indica el nombre del proyecto y el número de grupo.

Diseño de hardware final

El diseño y la fabricación de la placa PCB se llevaron a cabo conforme a las especificaciones previamente establecidas en el esquemático eléctrico. Durante la etapa de ruteo se realizaron ajustes mínimos, principalmente vinculados a la correcta

distribución de la alimentación y a la incorporación de puentes de GND con el objetivo de unificar las islas de masa que habían quedado eléctricamente aisladas de la bornera principal.

Durante el diseño se incorporó una región de conexión a tierra común, destinada a interconectar las referencias requeridas por los distintos periféricos del sistema, estableciendo un punto único de referencia eléctrica. Para mantener la continuidad de este plano de masa entre sectores que quedaban aislados debido al ruteo de las pistas, fue necesario implementar dos puentes conductores que permitieron unir dichas áreas y garantizar el correcto funcionamiento del circuito.

Asimismo, se efectuó un recorte en el lateral de la placa destinado a la conexión de la fuente de alimentación externa, con el objetivo de facilitar el acceso al conector del microcontrolador.

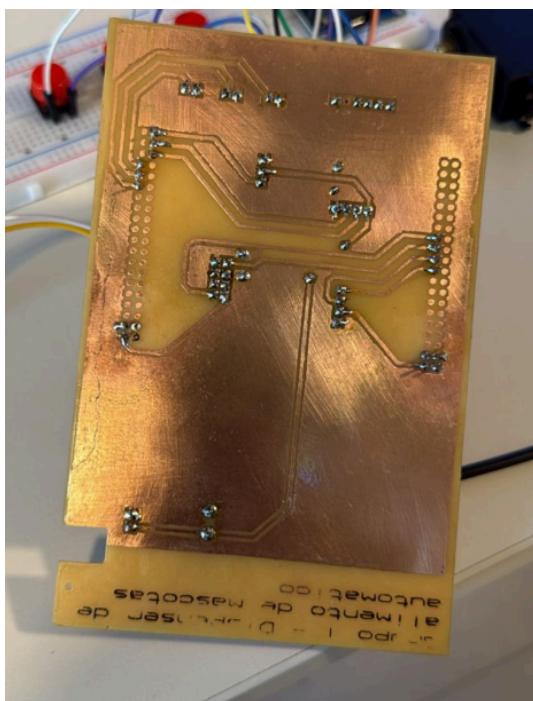


Imagen 10. PCB lado tracks

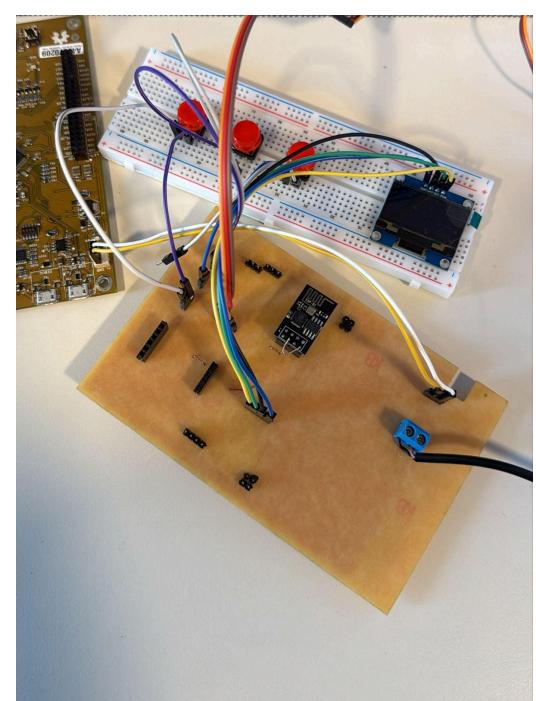


Imagen 11. PCB lado conector

Ensamblaje

Se incorporaron al modelo 3D tuercas de inserción térmica, las cuales fueron colocadas mediante un soldador para aplicar el calor necesario y permitir su correcta inserción en el material plástico. Este procedimiento asegura una fijación firme y

duradera, permitiendo que los tornillos y la varilla roscada se anclen adecuadamente a la pieza sin deteriorar la estructura del polímero.

Para permitir una medición precisa del peso del sistema, se diseñó y fabricó una base de madera que posibilita la correcta suspensión del dispensador, asegurando que la celda de carga trabaje únicamente bajo esfuerzo de tracción o compresión axial, sin interferencias estructurales.

Para ello, se realizó un corte en la sección central de una tabla, generando un espacio libre que permite suspender el conjunto sin que apoye sobre la base. De esta manera, la carga se transmite exclusivamente a la celda de carga, garantizando una medición correcta y evitando esfuerzos parásitos que puedan alterar la lectura.

Gestión de atascos de alimento en la hélice dispensadora

Durante las pruebas del sistema se detectó una limitación en el mecanismo de dispensado, específicamente el atascamiento del alimento en la hélice dispensadora. Para mitigar este inconveniente se realizaron distintos intentos de solución, como la variación de la velocidad de giro del servo mediante el control del PWM, la incorporación de una pieza adicional impresa en 3D a modo de embudo para dirigir el alimento hacia el centro y evitar que se acumule contra las paredes de la tolva, y la utilización de tres tipos diferentes de alimento balanceado. A pesar de estas pruebas, el problema persistió. Se observó que el sistema funciona correctamente cuando la hélice comienza a girar antes de que se agregue el alimento, o cuando se utiliza una cantidad mínima del mismo, lo que indica que el inconveniente está relacionado con la presión y distribución del material dentro de la tolva.

DISEÑO DEL FIRMWARE

Se eligió una arquitectura basada en eventos para el proyecto, ya que las interacciones del dispositivo dependen de eventos como la hora de dispensación, las mediciones de peso y las solicitudes de la app. Este enfoque permite que el microcontrolador se centre en procesar solo los eventos relevantes en el momento adecuado. Entre las ventajas de esta aproximación se incluyen la optimización del uso de recursos, ya que el microcontrolador solo se activa cuando es necesario, así como la

escalabilidad y un tiempo de respuesta rápido, al basarse en interrupciones y eventos.

Modularización

La modularización del proyecto tiene como objetivo organizar el sistema en componentes independientes y de responsabilidad clara. Esta estrategia facilita el desarrollo, mantenimiento y expansión del sistema al permitir que cada módulo se enfoque en una tarea específica, sin depender de otros componentes. Además, promueve la reutilización de código, la eficiencia en el trabajo en equipo y una estructura más clara y manejable, lo que hace que el proyecto sea más flexible y escalable a largo plazo.

1. Módulo de comunicación (ESP32 – WIFI)

El módulo de comunicación se encarga de gestionar la conexión Wi-Fi entre el microcontrolador y la aplicación móvil. Este módulo es responsable de establecer y mantener la conexión con la red Wi-Fi, así como de enviar y recibir datos entre el dispositivo y la app. Permite que el microcontrolador reciba configuraciones desde la app, como la hora de dispensación, y también le permite al usuario visualizar información como el porcentaje de llenado y el historial de dispensación. A través de este módulo, el sistema se comunica de manera remota, lo que proporciona flexibilidad y control desde cualquier lugar.

2. Módulo de control de dispensación (servomotor)

El módulo de control de dispensación es responsable de activar el servomotor que realiza la acción de dispensar la comida. Este módulo debe gestionar la cantidad de alimento a dispensar, basándose en la información recibida desde el sensor de peso y la configuración de la hora de dispensación. El servomotor, que fue intervenido para que realice giros completos, se controla para girar un número determinado de veces o para ajustar el ángulo de rotación según el peso calculado. Al implementar este módulo de manera independiente, se puede optimizar el control del servomotor y ajustar su comportamiento con precisión, sin interferir con otros módulos del sistema.

3. Módulo de sensor de peso (galga extensiometrífica)

El módulo de sensor de peso se encarga de medir el peso total del dispensador y el peso del alimento en el tanque. Utilizando la galga

extensiométrica, este módulo realiza lecturas de peso en tiempo real, permitiendo que el sistema calcule con precisión la cantidad de alimento que queda en el tanque y la cantidad a dispensar. Por ejemplo, si el depósito contiene 500gr y hay que dispensar 100gr, el motor debe detenerse cuando la medición del peso restante sea de 400gr. A medida que el peso cambia, el módulo genera eventos que son procesados por el microcontrolador, activando el servomotor cuando es necesario. Este módulo es clave para garantizar que el dispensador funcione de manera eficiente y precisa, entregando la cantidad correcta de comida en el momento adecuado.

4. Módulo de control de tiempo (RTC)

El módulo de control de tiempo gestiona la hora interna del dispositivo utilizando el RTC (reloj de tiempo real), que es un periférico del microcontrolador LPC4337. Este módulo se encarga de configurar y leer la hora actual, y de verificarla contra la hora de dispensación programada. Cuando la hora de dispensación llega, el módulo genera un evento que desencadena la acción de dispensar la comida. Además, se asegura de que el sistema esté sincronizado correctamente con el horario establecido, lo que permite que la dispensación se realice de manera automática y precisa sin necesidad de intervención manual.

5. Módulo de gestión de datos (APP)

El módulo de gestión de datos es el encargado de la interacción entre el microcontrolador y la aplicación móvil. A través de este módulo, el usuario puede configurar la hora de dispensación, consultar el porcentaje de llenado del tanque y visualizar el historial de dispensaciones. También permite que el microcontrolador reciba comandos desde la app, como cambios en la configuración del sistema. Este módulo facilita la administración remota del dispensador, brindando al usuario acceso completo y control del dispositivo, incluso si no está físicamente cerca de él.

Estructura de archivos

- examples/c/sapi
 - /main
 - Programa principal, inicialización y acceso al loop principal.
 - /ui
 - Interacción total con la pantalla y la máquina de estados.
 - /balanza/hx711_example
 - Inicialización, tarado y medición con la balanza.
 - /pantalla
 - Lógica para escribir en pantalla.
 - /servo
 - Funcionamiento para el movimiento del servomotor.
 - i2c/oled_sh1106
 - Librería para el funcionamiento de la pantalla, inicialización, llenado y limpieza.
 - esp8266/at_commands
 - Librería encargada de conectar el ESP a internet y recibir peticiones HTTP.

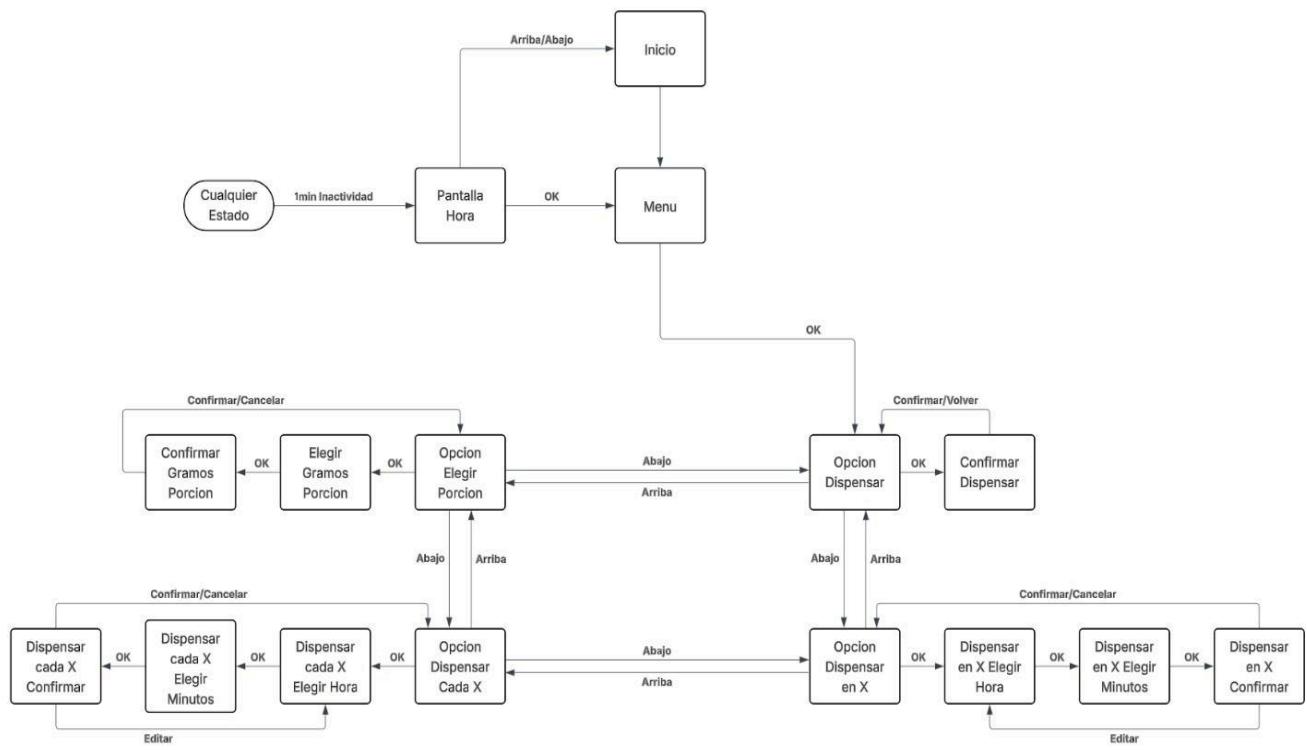


Figura 3. Diagrama de flujo de la MEF

Aplicación

Estructura de archivos de la APP:

Consta de un proyecto con varios archivos, dentro de los modificables se encuentra el directorio lib/ donde se encuentra el main de la app, y luego se separa en subdirectorios para organizar de esta manera:

- /apptaller1/lib
 - /screens
 - Posee las interfaces de las pantallas
 - /services
 - Capa de servicio, actúa como puente entre app y microcontrolador.
 - main.dart
 - Programa principal que arma la pantalla home.

Diagrama de flujo APP

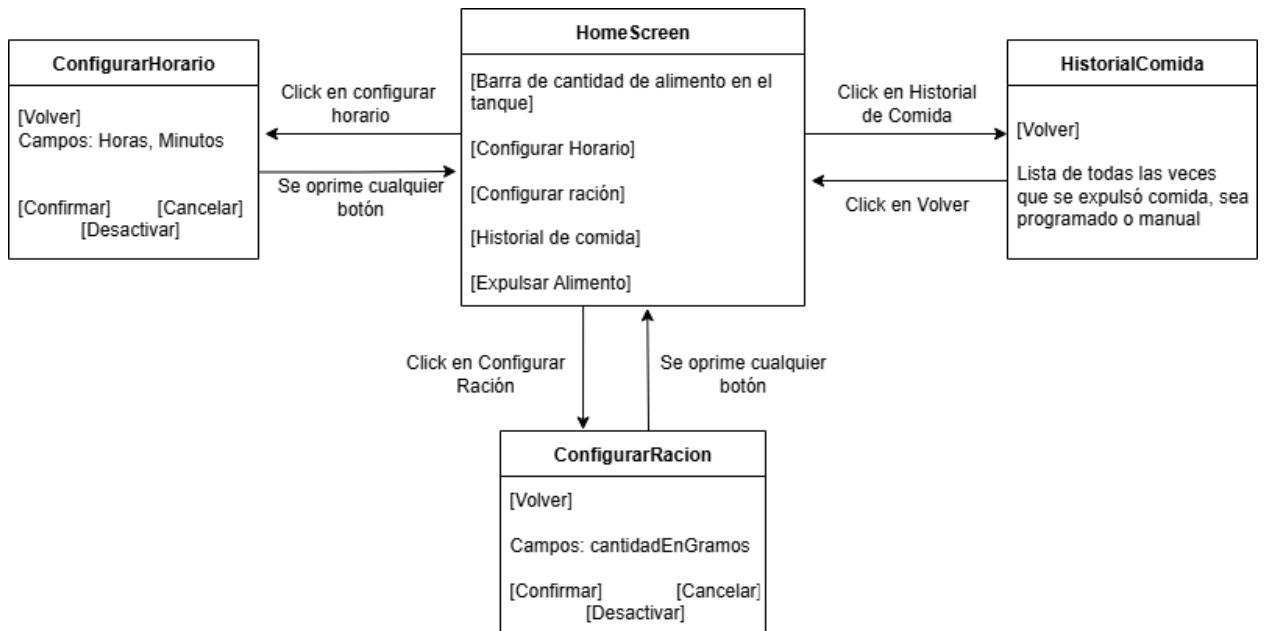


Figura 4. Diagrama de flujo de la app

Prototipo final de la APP

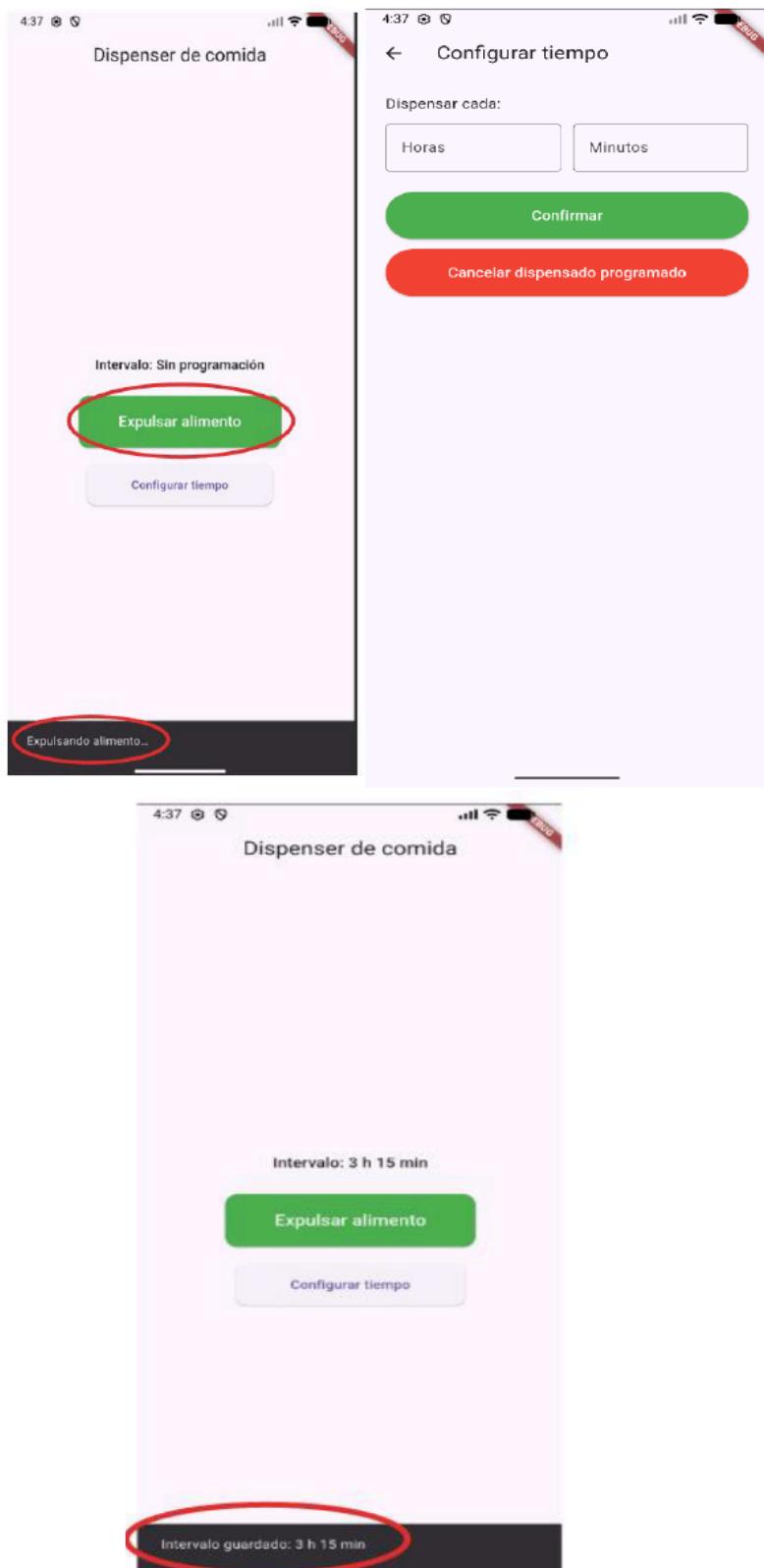


Imagen 12. Prototipo inicial de la APP

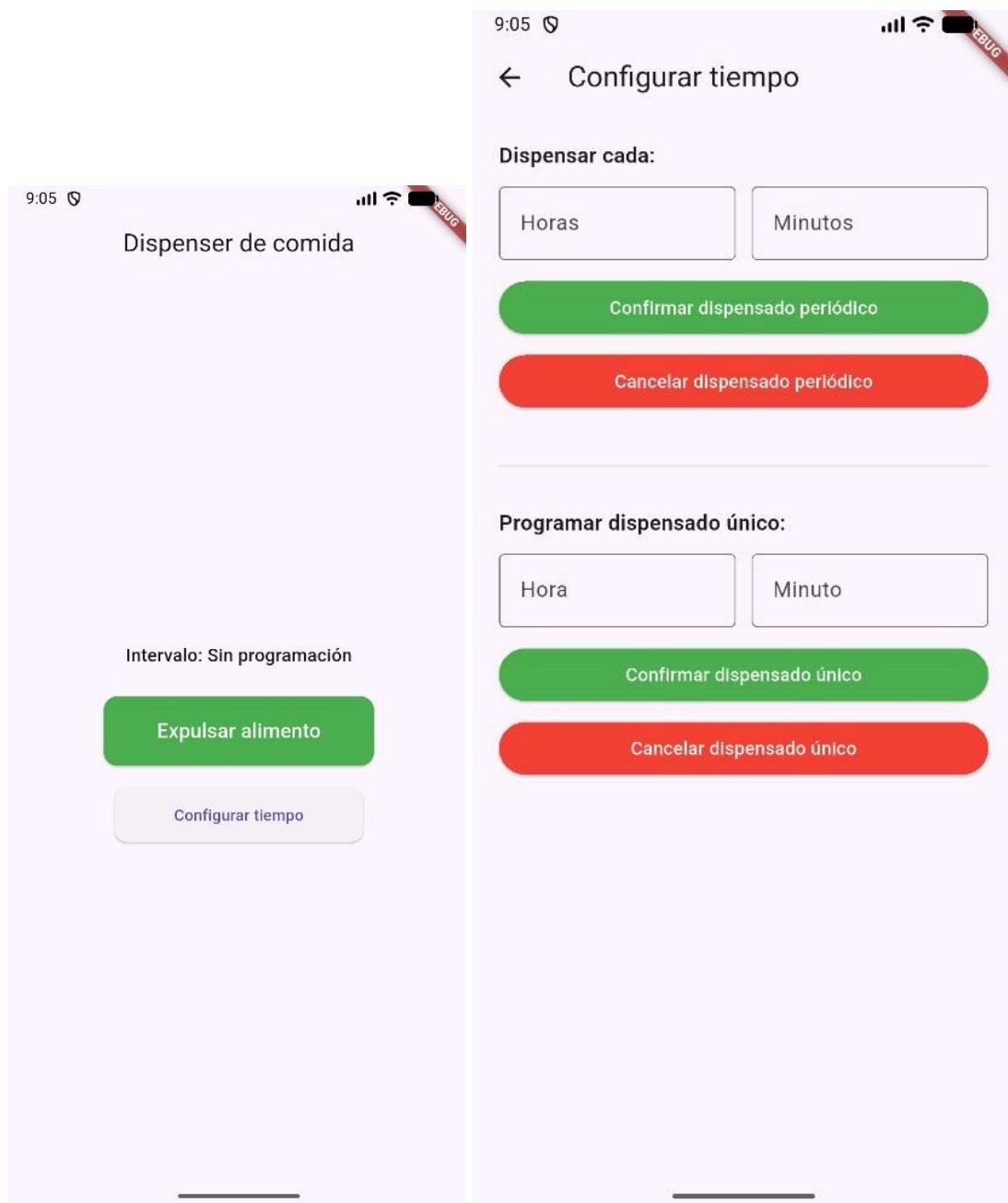


Imagen 13. Prototipo final de la APP

DESARROLLO DE SOFTWARE

El desarrollo del firmware se inició utilizando como base un archivo de ejemplo proporcionado por la cátedra, `uartBridge_ESP8266.c`. Este archivo fue diseñado originalmente para actuar como un simple "puente" UART manual, reenviando datos entre el monitor serie (UART_USB) y el módulo Wi-Fi (UART_232).

Avances de desarrollo (firmware y conectividad)

Inicialmente, se exploró la configuración del ESP8266 como estación/cliente, lo cual requería que el dispositivo se conectara a una red Wi-Fi existente. Sin embargo, para garantizar una demostración independiente a la red de la facultad, se redefinió la arquitectura.

El sistema ahora opera de forma autónoma en modo punto de acceso (Access Point). En esta configuración, el ESP8266 crea su propia red Wi-Fi, una tarea que se logra mediante la secuencia de comandos que se muestra en la *tabla 1*.

En este modo, el módulo adopta una dirección IP estática (192.168.4.1) y actúa como un servidor, asignando direcciones IP a los clientes que se conectan (como la PC o la app). La conexión de esta red local se validó inicialmente de forma manual a través de un comando ping desde la consola de comandos de una PC conectada.

Una vez validada la conectividad manual, el logro principal fue automatizar esta secuencia de configuración en el firmware de la EDU-CIAA. Se desarrolló un código C que ejecuta la serie de comandos AT uno tras otro. Para asegurar la fiabilidad del proceso, se implementó una función de control: para que cada comando se tome como exitoso y se envíe el siguiente, esta función verifica que la respuesta del ESP8266 sea válida (ej. "OK" o "ready").

Existe un problema a trabajar a futuro: El ESP presenta timeouts aleatorios o reinicios durante su uso junto con la app. Esto provoca que una vez que intenta reiniciarse, no valide correctamente los comandos, llevando a un error de conexión. Actualmente se cree que la causa del problema se debe a las conexiones mediante los cables jumper, siendo estos propensos a falsos contactos, dado que se ha comprobado empíricamente que desconectando todos los pines y volviéndose a conectar, asegurándolas firmemente, el problema se soluciona.

Avances de desarrollo (interfaz de usuario y aplicación)

Una vez se avanzó con el desarrollo del firmware, se trabajó en la aplicación cliente desarrollada en el lenguaje Flutter, usando el compilador Android Studio.

Inicialmente, se realizó una mejora significativa en la interfaz de usuario. Se desechó un primer diseño basado en ventanas y se migró a una arquitectura de página única (*Imagen 11*). Esto resulta en una experiencia de usuario más cómoda y simple.

Una vez establecido el diseño base, el trabajo se centró en la comunicación de la app con el microcontrolador. Esto se logró a través de peticiones HTTP a la IP del servidor. Hubo dos problemas clave:

1. **Verificación de Conexión (Chequeo de ping):** Al conectarnos a la red Wi-Fi del ESP, el navegador nos indicaba que la red "No tiene conexión a Internet" (lo que al principio se interpretó como un error). Este comportamiento resulta ser normal y esperado, pero un chequeo de conexión genérico en la app (que buscará acceso a Internet) fallaría. Para distinguir entre la advertencia normal de "sin internet" y un fallo real de conexión con el ESP, se implementó la función que envía una petición GET /ping directamente a la IP del microcontrolador. Para que esto funcionara, el bucle del firmware fue modificado para reconocer GET /ping y devolver una respuesta HTTP/1.1 200 OK. De esta forma, cuando la app recibe el statusCode == 200, sabe con certeza que está conectada al dispensador.
2. **Error de conexión con el navegador:** Inicialmente, se intentó comunicar la aplicación web de Flutter, ejecutándose en localhost (desde un navegador), con el ESP en su propia red. Esta comunicación no fue posible; se detectó que las peticiones HTTP nunca llegaban al microcontrolador y, por lo tanto, la aplicación no recibía ninguna respuesta. Se llegó a la conclusión de que este fallo se debía a las políticas de seguridad propias del navegador (CORS), que bloquean las peticiones de "origen cruzado" (de localhost a una IP de la red local). Para superar esta limitación y poder probarla, se optó por compilar la aplicación como un ejecutable nativo de Windows (más adelante se deberá hacer para Android/iOS). Al no estar sujeta a las restricciones de un navegador, la aplicación nativa pudo enviar las peticiones a 192.168.4.1 sin restricción, validando el método de comunicación y respondiendo correctamente.

Con la conexión ya estable, la app puede enviar comandos específicos. Por ejemplo, al presionar el botón "Dispensar Ahora", se envía una operación GET /dispensar. El microcontrolador, que está constantemente esperando estas peticiones, detecta este comando y responde con un 200 OK, permitiendo a la app confirmar al usuario que la operación fue exitosa.

El próximo paso sería vincular estas peticiones ya funcionales con las acciones físicas reales correspondientes, como implementar la lógica para mover el servomotor cuando se recibe el comando GET /dispensar y no sean una simple respuesta por consola.

Función	Comando	Descripción
1) Test de conexión	AT	Nos aseguramos de que la comunicación UART funciona si nos devuelve un 'OK'
2) Reset	AT+RST	Se borra cualquier configuración de conexión anterior.
3) Configurar modo	AT+CWMODE=2	Configura al ESP8266 en modo 2 (punto de acceso)
4) Configurar red	AT+CWSAP="[" + nombre_red + "]", "[contraseña]", "[canal]", "[encriptado]"	Crea la red con los datos ingresados
5) Habilitar múltiples conexiones	AT+CIPMUX=1	Permite varias conexiones. Obligatorio para levantar un servidor
6) Iniciar servidor	AT+CIPSERVER=1,80	Abre el puerto 80, se queda esperando peticiones HTTP provenientes de la app

Tabla 1. Comandos AT

Diseño de software final

El software final del dispensador se diseñó con una arquitectura modular y orientada a eventos, pensada para operar de forma confiable en un sistema embebido y con múltiples periféricos. La aplicación se organiza en capas funcionales separadas: drivers de hardware (GPIO/UART/RTC/OLED/HX711/servo), servicios (servidor HTTP, scheduler de dispensados, lectura y filtrado de peso), y lógica de aplicación/UI (máquina de estados del menú y configuración).

La modularización se refleja en componentes claros: el módulo de comunicación Wi-Fi encapsula el intercambio con el ESP8266 mediante comandos AT, incluyendo un parser de tramas para reconstruir requests HTTP, debido a que las requests llegaban en varios mensajes dependiendo de que tan larga sea la request, y una rutina de respuesta mínima HTTP 200 OK. Sobre eso se implementa un “router” de endpoints (/ping, /dispensar, /programar, /cada) que traduce parámetros de la URL a acciones internas. En paralelo, la interfaz local se implementa como una FSM (finite state machine) con estados de inicio, menú, pantalla de hora y pantallas de edición/confirmación para los modos programados, más la opción de dispensado manual. El manejo de botones se resuelve con antirrebote por software (ventana temporal + detección de flanco de presión), lo que separa la lectura directa del GPIO de los eventos utilizados por la FSM.

Para la planificación de dispensados, el sistema combina dos algoritmos: un disparo “una vez” basado en comparación contra el RTC (hora/minuto) con un flag anti-repetición para evitar re-disparos dentro del mismo minuto, y un modo repetitivo “cada X tiempo” basado en tickRead() con el cálculo del próximo vencimiento y una comparación ante un desborde de ticks. Ambas opciones llegan a una rutina de activación que actualiza la pantalla y activa la dispensación, manteniendo una sola forma de dispensar.

El subsistema de balanza se finalizó con una versión que lee 24 bits raw con extensión de signo, espera la señal de "datos listos" y usa un filtro de mediana de 5 muestras para eliminar valores atípicos y ruido. La conversión a gramos se realiza con una calibración (tara y factor de escala) para permitir medición continua sin recalibrar en cada arranque. Sobre ese servicio se apoya el algoritmo de dispensado: el servo se controla mediante generación de pulsos PWM por software y se opera en “pasos” (girar en pequeñas ventanas), midiendo el peso entre pasos para decidir el corte. Este método

emplea un control de lazo abierto basado en un umbral (objetivo de porción en gramos) con condiciones de seguridad como timeout.

En conjunto, la arquitectura final prioriza bajo acoplamiento (cada módulo conoce lo mínimo de los demás) y tolerancia a fallas (timeouts y filtros para sensores/serial). El resultado es un software donde el flujo principal ejecuta continuamente el loop de servidor, scheduler y UI, mientras que el hardware se administra a través de drivers pequeños y servicios específicos, manteniendo el comportamiento del sistema predecible y fácil de depurar.

En cuanto a la app, se logró vincular la misma con el EDU-CIAA, de esta manera teniendo una interfaz que permite establecer el horario programado de dispensado, o de dispensado cada determinado tiempo a través de solicitudes HTTP, así como también se pueden cancelar ambas opciones. Cada vez que la app envía un comando al dispensador, se despliega un mensaje en ambas pantallas expresando que la comunicación y su función fue completada con éxito. El único requisito para que todo esto pueda llevarse a cabo es que el celular esté conectado al WiFi brindado por el ESP8266.

7. ENSAYOS Y MEDICIONES

Para la verificación inicial del sistema una vez construida la PCB, se realizaron ensayos eléctricos utilizando un multímetro en modo prueba de continuidad. Estas mediciones permitieron comprobar la ausencia de cortocircuitos entre pistas adyacentes, el correcto funcionamiento de la región de conexión a tierra común y el correcto funcionamiento eléctrico de los puentes implementados para interconectar zonas aisladas del circuito. Asimismo, se verificó la correcta conexión entre los distintos conectores y sus correspondientes pines del microcontrolador, asegurando que las señales y la alimentación llegaran adecuadamente a cada periférico antes de proceder con las pruebas funcionales del sistema completo.

Tras el ensamblaje y la fabricación de la placa PCB, se procedió a reevaluar la estabilidad de la comunicación inalámbrica. Anteriormente, la conexión del módulo ESP8266/ESP32 a la red Wi-Fi en modo Access Point, realizada mediante cables jumper directamente a la EDU-CIAA, presentaba inestabilidad, errores de conexión intermitentes y reinicios inesperados del módulo, lo que dificultaba la comunicación fluida con la aplicación móvil. Con el módulo ESP ahora conectado directamente a la PCB, la cual

proporciona conexiones soldadas, se realizaron nuevas pruebas de conectividad. Efectivamente, se constató una mejora significativa en la fiabilidad de la conexión. La red se levanta y mantiene de forma estable, eliminando los timeouts y los fallos aleatorios que se atribuían a los falsos contactos de los cables. Esta validación fue crucial, ya que aseguró la base para la interacción remota del sistema y permitió la prueba del resto del software.

Tras la verificación exitosa de la PCB, surgieron desafíos durante las pruebas de integración del sistema. Un inconveniente se presentó al intentar obtener lecturas coherentes de la celda de carga a través del módulo HX711. Inicialmente, las mediciones arrojaban valores erráticos o sin sentido lógico, lo que impedía la calibración y el cálculo del peso. Se determinó que la causa principal era la conexión de la fuente de alimentación externa al circuito. Si bien el microcontrolador EDU-CIAA podía iniciar la lógica de control, la celda de carga y el servomotor, al ser alimentados por la PCB sin la fuente externa dedicada, el sistema no alcanzaba la estabilidad ni la potencia requerida para operar el sensor de peso y sus periféricos de forma fiable, lo cual fue resuelto al conectar la fuente externa, garantizando el voltaje y la corriente necesarios para todo el subsistema de sensado y actuación.

A pesar de los desafíos iniciales con la celda de carga y la necesidad de una fuente externa, la integración del resto de los periféricos se llevó a cabo sin mayores inconvenientes. Esto se debió a que las librerías de drivers provistas por la cátedra para el microcontrolador demostraron ser suficientes para la rápida puesta en funcionamiento de componentes clave como el servomotor, el display y los botones. La disponibilidad de estas herramientas facilitó enormemente la etapa de desarrollo, permitiendo al equipo concentrarse simplemente en adaptar estas librerías, en la lógica del sistema y la comunicación inalámbrica, en lugar de dedicar tiempo significativo a la escritura y depuración de drivers de bajo nivel.

La última etapa del diseño y ensamblaje fue respecto al montaje final del sistema. Se optó por utilizar una base de madera perforada para permitir que la celda de carga quedara suspendida en un punto central, garantizando que pudiera medir el peso del dispensador completo. Posteriormente, el cuerpo del dispensador, fue acoplado directamente sobre la celda de carga. El resto de los componentes electrónicos, se dispusieron junto a la base.

8. CONCLUSIONES

En función de los requerimientos y objetivos establecidos al inicio del proyecto, se ha realizado una evaluación del prototipo final, obteniendo los siguientes resultados:

Cumplimiento de Objetivos

- **Objetivos Primarios:** Se ha alcanzado un grado de cumplimiento total de los objetivos primarios propuestos. El sistema cuenta con un control basado en el microcontrolador EDU-CIAA que gestiona con éxito los tres modos de operación. Se integró un mecanismo de dispensado funcional y un temporizador programable mediante RTC que satisface los horarios dictados por el usuario. Asimismo, la aplicación móvil permite la configuración de intervalos y la activación manual del dispositivo de manera efectiva.
- **Objetivos Secundarios:** En cuanto a las mejoras propuestas, se logró integrar satisfactoriamente el sistema de sensado de peso mediante la galga extensiométrica para el monitoreo del depósito. No obstante, ha quedado pendiente de implementación la adición de un sistema de registro histórico detallado del consumo de alimento diario.

Evaluación de Requerimientos

- **Requerimientos Funcionales:** Se cumplieron satisfactoriamente casi todos los requerimientos operativos, incluyendo la capacidad del depósito para contener alimento, el suministro automático por horario o intervalos, y la definición del peso de las porciones en gramos. Sin embargo, el requerimiento de informar en tiempo real tanto en la aplicación móvil como en la pantalla física la cantidad de comida disponible (en gramos y porciones) no fue completado en su totalidad, quedando esta funcionalidad como una mejora para futuras versiones.
- **Requerimientos No Funcionales:** Se cumplió estrictamente con el uso de la plataforma EDU-CIAA. El dispositivo presenta dimensiones y peso aptos para el entorno doméstico, con un diseño amigable para la interacción con mascotas. Finalmente, el desarrollo se ajustó de manera exitosa al cronograma de la materia.

Conclusión General

La experiencia de la materia, centrada en el desarrollo completo de un proyecto embebido, constituyó una gran oportunidad de aprendizaje. Más allá de la ejecución técnica, el curso exigió la integración de conceptos críticos como el diseño de hardware (PCB), el desarrollo de firmware con diversos periféricos (servomotores, sensores y displays) y la gestión rigurosa de requerimientos y tiempo. Este enfoque permitió comprender la interconexión entre las distintas capas de un sistema embebido, desde la electrónica de bajo nivel hasta la lógica de control de alto nivel. Todas las metodologías de ingeniería aplicadas nos preparan sólidamente para el campo profesional.

El proyecto constituye un prototipo funcional sólido que aborda la problemática de la alimentación irregular mediante la aplicación de herramientas tecnológicas, cumpliendo parcialmente con las funcionalidades planteadas en los objetivos iniciales. Los desafíos técnicos superados durante la integración de hardware y software, especialmente en la estabilización de la comunicación inalámbrica y la calibración de la balanza, han brindado una experiencia de aprendizaje integral en el área de sistemas embebidos.

Si bien el sistema logra ejecutar correctamente sus funciones principales, se identificó como limitación el atascamiento del alimento en la hélice dispensadora, aspecto que queda como línea de mejora para futuras iteraciones del proyecto.

Trabajos a futuro

Como resultado del análisis del prototipo final y de los requerimientos que no alcanzaron un cumplimiento total, se plantean las siguientes líneas de desarrollo para futuras iteraciones del proyecto:

- **Monitoreo de stock en tiempo real:** Implementar la lógica necesaria en el firmware para transmitir el peso medido por la celda de carga tanto a la pantalla como a la interfaz móvil, permitiendo al usuario conocer los gramos de alimento disponibles.
- **Rediseñar el sistema de dispensado de alimento:** Proponer una nueva arquitectura del sistema para optimizar el flujo del alimento y minimizar los atascos observados durante el funcionamiento. Entre las posibles alternativas analizadas, que no pudieron ser implementadas en el proyecto, incluyen la

modificación del embudo para que el alimento fluya adecuadamente hacia la hélice o reducir el tamaño de las aspas de la hélice.

- **Registro de consumo histórico:** Desarrollar un sistema para almacenar el historial de dispensaciones o un registro por día, facilitando un seguimiento detallado del consumo de la mascota a lo largo del tiempo.
- **Compilación multiplataforma:** Finalizar la compilación de la aplicación para sistemas operativos iOS, eliminando la dependencia actual de la ejecución sobre Android.

División de tareas final

- Martín Graziani:
 - Modificación del servomotor para permitir su funcionamiento en modo de rotación continua, diseño del esquemático eléctrico y de la placa PCB del sistema, corrección de pistas de la PCB, realización de pruebas de continuidad eléctrica y verificación de ausencia de cortocircuitos, diseño y armado del soporte mecánico para la celda de carga, compra de los componentes y materiales necesarios para el proyecto, elaboración de los informes, confección del presupuesto general del sistema, corte y preparación de la varilla para la estructura mecánica del dispositivo, adquisición y acondicionamiento del modelo 3D del sistema.
- Manuel Reyes:
 - Diseño del esquemático y de la PCB del sistema, soldadura de pines y conectores en la placa, verificación de continuidad eléctrica y corrección de pistas, ensamblado del dispensador y colocación de roscas de inserción térmica, acondicionamiento de la hélice de dosificación, selección de componentes electrónicos y mecánicos, planificación de modificaciones estructurales y funcionales del dispensador, y planificación del modo de montaje y funcionamiento mecánico del sistema.
- Joaquín Rodriguez:
 - Diseño e implementación del software embebido del sistema, incluyendo la arquitectura general de la aplicación y su modularización. Desarrollo de la interfaz de usuario con menú navegable mediante botones físicos y visualización en pantalla OLED. Desarrollo del sistema de planificación de dispensados, incorporando modos de dispensado manual, programado por horario mediante RTC y dispensado automático a intervalos

configurables. Implementación del módulo de medición de peso mediante la galga HX711, filtrando de lecturas, calibración y conversión a gramos. Integración del control del servomotor con la balanza para permitir el dispensado por porción, deteniendo automáticamente el suministro al alcanzar el peso objetivo.

- Juan Martín Castro Bustos:

- Responsable del diseño e implementación del firmware para el microcontrolador y sus periféricos. El desarrollo se realizó mediante módulos independientes para facilitar la integración final del sistema. Esto incluyó la comunicación con el módulo ESP, la adaptación del driver para el display OLED y el control del servo modificado. Asimismo, se implementó la lógica del sensor de peso, el filtrado de datos y la calibración. Además de integrar todos los módulos, se colaboró en la aplicación móvil desarrollando la lógica de comunicación HTTP con el dispensador.

Integrante	Horas aproximadas acumuladas al (23/02/2026)
Castro Bustos Juan Martín	88hs
Graziani Alejandro Martín	86hs
Reyes Manuel	88hs
Rodríguez Joaquín	87hs

Las horas se definen con la cantidad de asistencias a clases (2hs c/u) sumando una aproximación de horas extracurriculares.

Cronograma

Cronograma Tentativo			
semana	Fecha	ENTREGA DE INFORMES	ACTIVIDADES DE PROYECTO
1	lunes 18/08/25		Formación de grupos Elección de proyecto Análisis de componentes necesarios y sus costos asociados
	jueves 21/08/25		
2	lunes 25/08/25		Análisis de componentes necesarios y sus costos asociados
	jueves 28/08/25		
3	lunes 1/9/2025		Elaboración de informe inicial Diseño de piezas Análisis de requerimientos Distribución preliminar de tareas Presentación de los proyectos grupales propuestos
	jueves 04/09/25		
4	lunes 8/9/2025	ENTREGA INFORME INICIAL	Elaboración de informe inicial Diseño de piezas Análisis de requerimientos Distribución preliminar de tareas Presentación de los proyectos grupales propuestos
	jueves 11/9/2025		
5	lunes 15/9/2025		Desarrollo y diseño Hard-Soft Desarrollo informe avance 1
	jueves 18/09/25		
6	lunes 22/9/2025		HW: diseño del esquemático y estudio de los componentes (precios, alternativas, funcionamiento) SW: prototipo de app móvil
	jueves 25/9/2025		
7	lunes 29/9/2025		Consultas y entrega de informes
	jueves 02/10/2025	VENCIMIENTO INFORME DE AVANCE 1	
8	lunes 06/10/2025		Corrección de los diseños Adquisición de materiales
	jueves 09/10/2025		
9	lunes 13/10/2025	Feriado Respeto a la diversidad cultural	
	jueves 16/10/2025		Implementación - prototipado Desarrollo informe avance 2 HW: Diseño de la PCB SW: nuevo prototipo de la app, comunicación aplicación-microcontrolador exitosa, lógica del sistema para responder a
10	lunes 20/10/2025		
	jueves 23/10/2025		

11	lunes 27/10/2025		las peticiones HTTP
	jueves 30/10/2025		
12	lunes 03/11/2025	VENCIMIENTO <i>INFORME DE AVANCE</i> 2	Consultas y entrega de informes
	jueves 06/11/2025		Corrección de los diseños
13	lunes 10/11/2025		
	jueves 13/11/2025		
14	lunes 17/11/2025		Corrección de los diseños y fabricación
	jueves 20/11/2025		
15	lunes 24/11/2025	Feriado: Día de la soberanía Nacional	
	jueves 27/11/2025		Corrección de los diseños y fabricación
16	lunes 01/12/2025		
	jueves 4/12/2025		Integración, test y validación
17	lunes 08/12/2025	Feriado: Día de la inmaculada concepción de la Virgen María	
	jueves 11/12/2025		Integración, test y validación
18	lunes 15/12/2025		
	jueves 18/12/2025		Documentación Final

RECESO DE VERANO - 2025

19	lunes 02/02/26		Documentación Final
	jueves 5/2/2026		
20	lunes 9/2/2025	Entrega de INFORMES.	Entrega de informe
	jueves 12/2/2025		
21	lunes 16/2/2025	Feriado : Carnaval	
	jueves 19/2/2025	FECHA DE PRESENTACIONES DE PROYECTOS	
22	lunes 23/2/2025	FECHA DE PRESENTACIONES DE PROYECTOS	

BIBLIOGRAFÍA

Repositorio con archivos del proyecto:

[1] <https://github.com/rmanuelx5/Taller-de-proyecto-1>

Repositorio firmware

[2] https://github.com/epernia/firmware_v3

ANEXOS

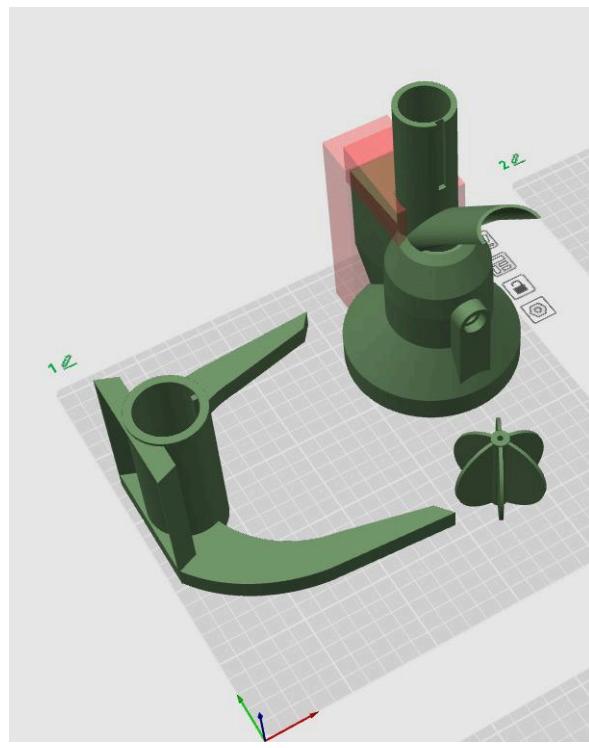


Figura A.1: Esquema para impresión 3D

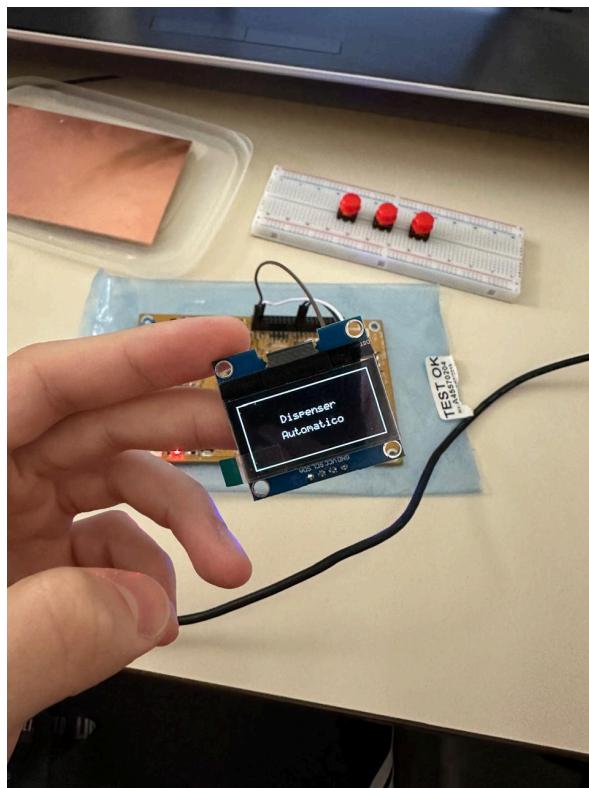


Figura A.2: Primer funcionamiento del display



Figura A.3: Modificando el servo para giros completos

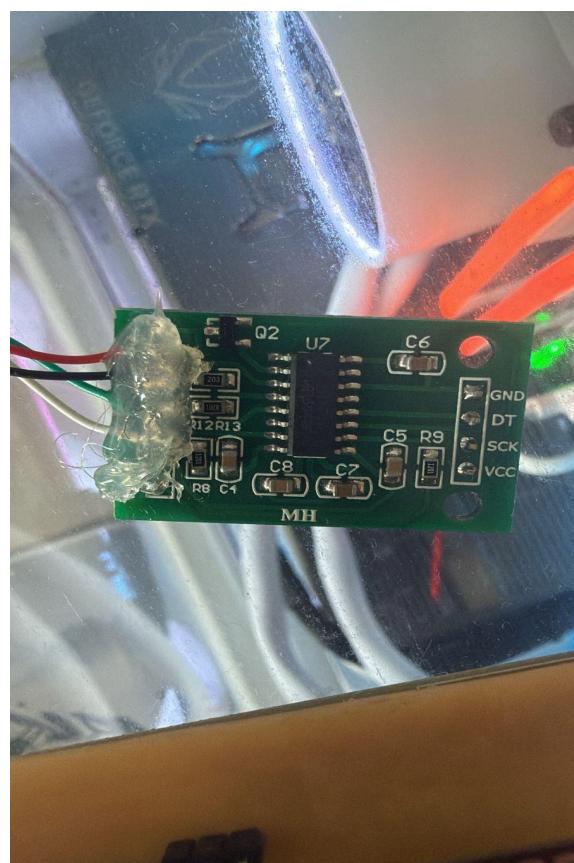


Figura A.4: Solución ante las reiteradas desconexiones

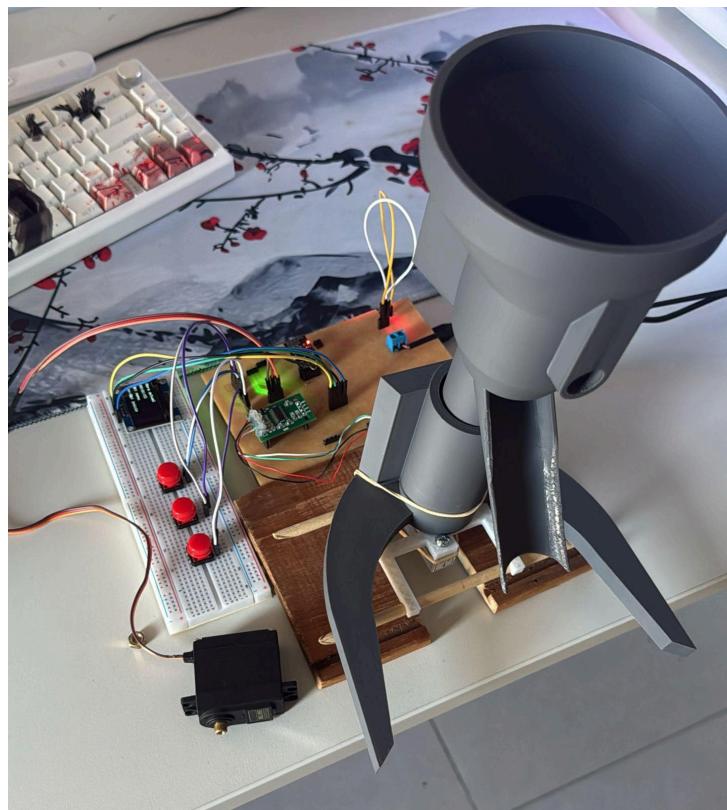


Figura A.5: Tara inicial con el dispenser encima

[Video del funcionamiento](#)